IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Graeme John PROUDLER	)	RE: Claim to Priority
Serial No.: 10/688,397	)	
Filed: 16 October 2003	)	Our Ref: B-5268 621375-8
For: "METHOD AND APPARATUS FOR MANAGING A HIERARCHY OF NODES"	)	Date: November 21, 2003

CLAIM TO PRIORITY UNDER 35 U.S.C. 119

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

[X] Applicants hereby make a right of priority claim under 35 U.S.C. 119 for the benefit of the filing date(s) of the following corresponding foreign application(s):

COUNTRY	FILING DATE	SERIAL NUMBER
GB	3 February 2003	0302410.6
GB	7 March 2003	0305225.5

[ ] A certified copy of each of the above-noted patent applications was filed with the Parent Application No._____.

[X] To support applicant's claim, certified copies of the above-identified foreign patent applications are enclosed herewith.

[ ] The priority document will be forwarded to the Patent Office when required or prior to issuance.

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first-class mail in an envelope addressed to the "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450", on November 21, 2003 by Ericca Long

Respectfully submitted,

Richard P. Berg
Attorney for Applicant
Reg. No. 28,145

LADAS & PARRY

10/688,597

PATENTS • DESIGNS • COPYRIGHT • TRADE MARKS
The Patent Office

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed K Behen

Dated 21 October 2003

10/688,397

Patents Form 1/77

Patents Act 1977
(Rule 16)

THE PATENT OFFICE
A
- 3 FEB 2003
RECEIVED BY FAX

The Patent Office

03FEB03 E781994-1 D01463
P01/7700 0.00-0302410.6

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

03 FEB 2003

The Patent Office
Cardiff Road
Newport
South Wales
NP10 8QQ

1. Your reference

2. Patent application number
(The Patent Office will fill in this part)
0302410.6

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*
Hewlett-Packard Development Company, L.P.
A Texas limited partnership
20555 S.H. 249
Houston, TX 77070

Patents ADP number *(if you know it)*
855788600 1

If the applicant is a corporate body, give the country/state of its incorporation
Texas, USA

4. Title of the invention
KEY HIERARCHY AND METHOD AND APPARATUS FOR HANDLING THE SAME

5. Name of your agent *(if you have one)*
Robert F. Squibbs
Hewlett-Packard Ltd. IP Section
Filton Road, Stoke Gifford
Bristol BS34 8QZ

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

Patents ADP number *(if you know it)*
7928187001

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))
Yes

Patents Form 1/77

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	-
Description	10 ✓
Claim(s)	3 ✓
Abstract	1 ✓
Drawing(s)	3 only ✓

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	
Request for preliminary examination and search *(Patents Form 9/77)*	
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	Fee Sheet ✓

11. I/We request the grant of a patent on the basis of this application.

Signature RF Squibbs Date

Robert F. Squibbs 3 / 2/2003

12. Name and daytime telephone number of person to contact in the United Kingdom Tony Judd Tel: 0117-312-8026

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 08459 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

Key Hierarchy and Method and Apparatus for Handling the Same

Field of the Invention

The present invention relates to a key hierarchy for use, for example, in protecting sensitive data used by a trusted platform; the present invention also relates to a method and apparatus for handling such a key hierarchy.

Background of the Invention

TCPA technology is specified in the TCPA specifications and described, for example, in the book "trusted computing platforms – tcpa technology in context"; Pearson (editor); Prentice Hall; ISBN 0-13-009220-7".

A trusted platform built according to today's TCPA specifications will incorporate a trusted platform subsystem typically comprising a Trusted Platform Module (TPM) in the form of a hardware chip separate from the main CPU, a Root of Trust for Measurement (RTM) formed by the first software to run during the boot process, and support software termed the Trusted platform Support Service (TSS) which performs various functions such as those necessary for communication with the rest of the platform. In a PC, the RTM will typically be founded upon BIOS instructions that cause the main platform processor to do RTM work; this set of instructions is called the Core Root of Trust for Measurement (CRTM) for convenience.

The RTM and associated measurement agents carry out integrity measurements (integrity metrics) on the platform at various stages and store the results in a measurement log in ordinary memory; however, a condensed summary is also stored in Platform Configuration Registers (PCRs) of the TPM.

In addition to the PCRs, the TPM comprises a processor and various cryptographic functions as well as memory for permanently holding secrets such as the private TPM endorsement key and the storage root key (SRK). With regard to the SRK, the TPM supports a Protected Storage mechanism in the form of a hierarchy (tree) of data objects the

root of which is the SRK; apart from the SRK that is permanently stored in the TPM (and not released from it), the tree can be stored outside of the TPM. Each intermediate node object in the tree is encrypted by a key in the node object above it in the tree (the parent node), all the way back to the SRK root node. Each key has an associated authorisation value that must be presented to the TPM (or, more accurately, used in a protocol that proves knowledge of the value without revealing the value) before the TPM permits the key to be used. Intermediate nodes in the tree will always be keys but leaf nodes can be arbitrary data (though frequently they will also be keys, such as symmetric keys for use by application processes in protecting bulk data). Keys in the tree can either be "non-migratable" meaning that the private key is only known to the TPM, or "migratable" meaning that there is no guarantee about the origin and use of the private key.

Access to keys in the key hierarchy (and thus to the data protected by the keys) can be made dependent on the current state of the platform as indicated by the values held in the PCRs. The relevant TCPA functions are "TPM_Seal" and TPM_Extend which enable a TPM to conceal decryption keys unless the value of current PCR is the same as stored PCR values. This sealing process ("Seal") enables enforcement of the software environment (PCRs) that must exist before data can be revealed, and simultaneously provides a method of concealing data (because the TPM releases a decryption key) until that environment exists. Seal is therefore an access control that depends on the *previous* state of a platform (represented in terms of PCRs). Seal permits the creator of data to dictate the software environment that must exist in a platform when the data is used

A trusted platform built according to today's TCPA specifications can be relied upon to store secrets, provide a platform identity, and reliably report the Operating System (OS) in a platform. Common operating systems cannot be relied upon to protect secrets once they have been revealed to the platform's software, nor to report on what has happened since the OS took control of the platform. This is because these OS's cannot be relied upon to reliably measure and store integrity metrics in the TPM's PCRs. It's not that common OSs can't measure and store integrity metrics, it's that they cannot protect themselves against subversion, so integrity metrics can't be trusted if they were stored after the OS was loaded. One consequence of the lack of trustworthiness of the OS is that the sealing

process described above has limited value in platforms with existing conventional OSs, because they cannot be relied upon to reliably measure and store integrity metrics in the TPM's PCRs.

Thus, current trusted platforms are limited because today's Core Root of Trust of Measurement (CRTM) is in a relatively unprotected BIOS chip, and because Operating Systems are insufficiently protected against subversion. Nevertheless, all functions that are provided by a TPM are fully protected, and if the platform's pre-boot environment (especially its BIOS Boot Block, acting as the TCPA Core Root of Trust for Measurement) is properly designed, all operations that executed before the OS can be faithfully recorded in the TPM's PCRs.

If the software on a platform is written to take advantage of the processing "rings" (levels of privilege) on processors (such as Intel processors), that software can use these rings to protect itself against subversion. A secure compartment-OS that is protected by virtue of these processing rings can dynamically create and destroy isolated software compartments. Applications then execute in isolated compartments, which protect the application from other applications, and visa versa. Each compartment provides and protects access to the secrets belonging to the application in the compartment. In servers, or peer-to-peer systems, trusted applications can talk to each other, whether they are in other compartments on the same platform or in compartments in other platforms. One important feature of this type of processing is that it prevents even the platform's administrator from violating the privacy of an application and data.

It is an object of the present invention to facilitate the use of protected processes in trusted platforms and, in particular, the release of keys to such processes. However, the present invention has wider application and is not to be limited in scope by the foregoing objective.

Summary of the Invention

The present invention is based in part on the observation that previous platform history is irrelevant for protected software provided that all traces of previous software have been unloaded or existing software is benign (such as a protected compartment OS). In these

cases, the operation of protected software is unaffected by software that previously executed on the platform. The decision to release secrets for use by protected software can therefore be made purely on the basis of knowing what protected software is about to be executed. So, if a TPM "knows" that protected software is about to be started, and is presented with a digest of that software, the TPM can safely release the secrets for that protected software.

In its broader aspects, the present invention is concerned with the mechanism used to ensure that only secrets associated with a protected process are released by the TPM. This mechanism involves the use of a "dynamic root key" that is associated with the protected process to be run, the key itself forming part of the key hierarchy stored in Protected Storage. When a protected process is run (or about to be run), the associated dynamic root key is installed in the TPM to act as the root of a hierarchy of (external) data objects instead of the SRK. Access to parts of the key hierarchy that require ascent from the dynamic root key is prohibited. To ensure that the process associated with a dynamic root key is the expected protected process, activation of the dynamic root key in the TPM preferably requires the presentation of an authorisation value that is, for example, a digest of the protected process. When the TPM loads a dynamic root key, the TPM unloads any existing keys (including any previous dynamic root key), invalidates any keys that might have been cached outside the TPM, and uses the new dynamic root key instead of the Storage Root Key. The TPM continues to use the new dynamic root key instead of the SRK until a new dynamic root key is loaded or until the TPM is restarted.

Once a dynamic root key has been loaded, it or its tree of data objects can be used to store keys that protect sensitive data belonging to the protected process.

Although the present invention has been outlined above in the context of the TCPA architecture, it is of broader application.

More particularly, according to one aspect of the present invention, there is provided a tree-structured key hierarchy with multiple nodes serving as root nodes dividing the hierarchy into different parts only accessible from corresponding root nodes.

According to another aspect of the present invention, there is provided processing apparatus comprising a key-handling unit for handling a tree-structured key hierarchy, the key-handling unit being arranged to treat a selected node of the hierarchy as the current root node such that those parts of the hierarchy that can only be reached by ascent from the current root node are inaccessible, the key-handling unit including an arrangement for changing the node of the hierarchy serving as said current root node.

The present invention also contemplates a method of managing a tree-structured hierarchy corresponding to that implemented by the foregoing apparatus.

Brief Description of the Drawings

Embodiments of the invention will now be described, by way of non-limiting example, with reference to the accompanying diagrammatic drawings, in which:

- **Figure 1** is a diagram of key hierarchy associated with a Trusted Platform Module;
- **Figure 2** is a diagram indicating the contents of a dynamic root key object of the Figure 1 key hierarchy;
- **Figure 3** is a diagram illustrating trusted platform elements involved in starting a protected process; and
- **Figure 4** is a diagram illustrating certain stages in the activation of a dynamic root key associated with the Figure 3 protected process.

Best Mode of Carrying Out the Invention

Figure 1 illustrates a trusted Platform Module (TPM) 10 with its normal Protected Storage data-object hierarchy 12 (also referred to below as a key hierarchy). The TPM's Storage Root Key (SRK) 11 resides permanently inside the TPM 10. The SRK 11 is used to encrypt ("wrap") keys K1-1, K1-2, K1-3 etc. that form the next level of the hierarchy. Key K1-1, which in this case is a non-migratable key, itself wraps further keys K2-1 , K2-2, etc. The hatched outer annulus around each key in the Figure 1 key hierarchy 12 is a graphical indication that each key is wrapped (encrypted). A key in the hierarchy 12 can only be decrypted by the TPM 10 upon presentation to the latter of authorizations in respect of the ancestor keys in the hierarchy.

In the present example, Key K2-1 is a "dynamic root key object" 16. The dynamic root key 16 can be the child of any non-migratable node in the key hierarchy 12; for example, the dynamic root key 16 could be a child of the SRK 11 rather than of key K1-1 as illustrated. As will be described below, the dynamic root key 16 acts as the root of a hierarchy of (external) data objects instead of the SRK, in respect of an associated protected process.

The standard TCPA definition of a key is:

```
typedef struct tdTCPA_KEY{
        TCPA_VERSION ver;
        TCPA_KEY_USAGE keyUsage;
        TCPA_KEY_FLAGS keyFlags;
        TCPA_AUTH_DATA_USAGE authDataUsage;
        TCPA_KEY_PARMS algorithmParms;
        UINT32 PCRInfoSize;
        BYTE* PCRInfo;
        TCPA_STORE_PUBKEY    pubKey;
        UINT32 encSize;
        [size_is(encData)] BYTE* encData;
        } TCPA_KEY;
```

In fact, only the data in "encData" is encrypted (wrapped) by the key above in the key hierarchy, the other data being in cleartext (though any changes are detectable as a digest of that data forms part of the encrypted data).

The keyFlags variable is a structure that indicates Boolean properties of the key, currently redirection, migratable, and volatileKey.

In order to support dynamic root keys, according to the present embodiment, a new key flag DRK (dynamicRootKey) is added to KeyFlags. If the TPM 10 receives a "create key" command with a TCPA_KEY structure where the DRK flag is set, the TPM 10 automatically creates a dynamic root key without any change to the existing key creation functionality of the TPM.

Figure 2 illustrates the data structure of a dynamic root key, such as key 16, created by the TPM 10; as can be seen, the keyflags data 18 includes the DRK flag 19 which is in its 'set' state. The dynamic root key 16 also includes an authorisation value 17 which forms part of the encrypted data in "encData"; in the present case where the key 16 is a dynamic root key, the authorisation value 17 is the digest of a protected process (that is, a process that either protects itself, or is protected by another entity, from subversion). The digest is creating by processing with a hash algorithm the digital data that represents the protected process.

Rather than the TPM 10 creating a dynamic root key on the basis of the state of the DRK flag in a TCPA_KEY structure associated with a "create key" command, a separate flag could be added to the command itself to indicate that the new key is a dynamic root key.

Whenever the TPM loads a key for which the DRK flag 19 is set, it knows that the key is a dynamic root key and proceeds accordingly, as described below.

Figure 3 illustrates the trusted platform elements involved in starting a protected process 25 such as may be formed by a secure-compartment OS. The TPM 10 of the trusted platform is shown in its condition prior to the activation of the protected process 25 with its SRK 11 forming the root of the available key hierarchy 12 held in Protected Storage 23 that is physically stored in normal memory 22.

A Root of Trust (RT) 20 is responsible for activating the protected process 25 by initiating its execution by the platform's main processor 27 and by causing the TPM 10 to unlock the dynamic root key and enable its use, for example, to access keys that depend from it in the key hierarchy or to wrap and unwrap data presented to the TPM. The RT 20 is analogous to the RTM of a trusted platform and may, for example, be a hardware device or the platform as a whole running trusted software. The RT 20 is able to communicate with the TPM 10 in a manner that enables the TPM to believe that the communication came from the RT 20 (for example, the communication can be passed over a dedicated virtual or physical channel 21).

The various stages involved in the installation of the dynamic root key 16 (and any dependent object hierarchy) that is associated with the protected process 25 are described below with certain of these stages being illustrated in Figure 4.

1. At switch-on of the trusted platform, the TPM 10 (always) has the SRK 11 at the root of the available key hierarchy (see Figure 4).
2. At some point, the RT 20 determines that the protected process 25 associated with the dynamic rootkey16 is the next process to execute and either has exclusive access to the TPM 10 (because the RT 20 has, for example, cleared all other processes from memory), or any existing protected processes will not abuse information obtained from the TPM (for example, the RT 20 may itself be a secure-compartment OS wishing to start process 25 in a compartment).
3. The RT 20 computes a digest of the protected process 25.
4. The key object K1-1 is loaded (using TPM_Loadkey) into the TPM 10 with authorisation to use the SRK 11; the load command and the related authorisation may come from the RT 20 or another element of the platform triggered by the RT 20.
5. The TPM 10 decrypts the key object K1-1 using the SRK's private key and obtains the key inside the K1 key object.
6. The key object K2-1 (the dynamic root key object 16) is loaded into the TPM 10 with authorisation to use the unwrapped key K1-1; again, the load command and the related authorisation may come from the RT 20 or another element of the platform triggered by the RT 20 .
7. The TPM 10 checks that unwrapped key K1-1 is a normal key by checking its DRK flag bit in the KeyFlags structure, and uses the unwrapped private key of key object K1-1 to decrypt key object K2-1 and obtain the key inside the dynamic root key object 16.
8. The RT 20 presents a TPM_ActivateDynamicRoot command to the TPM 10 with authorisation to use the key inside the key object K2-1, this authorisation being the digest of the protected process 25 associated with the dynamic root key 16 (see Figure 4).

9. The TPM 10 verifies that the TPM_ActivateDynamicRoot command came from the RT 20 (for example, by checking the channel on which the command was received), and that the key inside the key object K2-1 is a dynamic root key (by checking its DRK flag bit in the KeyFlags structure).

10. The TPM 10 deactivates, but does not unload or discard, the SRK 11 and uses the dynamic root key 16 from the K2-1 key object as if it were the SRK (see Figure 4). In other words, the key 16 forms the root of a hierarchy of data objects, and its private key never leaves the TPM in plaintext form. When the TPM 10 loads the dynamic root key 16, the TPM unloads any existing keys including any previous dynamic root key (the SRK 11 is, however, retained in the TPM), and invalidates any keys that might have been cached outside the TPM 10 in encrypted form

Thereafter, the dynamic root key 16 acts as the root of a hierarchy of (external) data objects instead of the SRK 11, in respect of the associated protected process 25. One consequence of this is that it is no longer possible to go higher up the key hierarchy 12 than the dynamic root key 16 whilst the latter is activated. The TPM 10 continues to use the dynamic root key 16 instead of the SRK 11 until either a new dynamic root key is loaded and activated or until the TPM 10 is restarted.

With the dynamic root key 16 installed, the protected process 25 can be run and can access data protected by the key 16 or a key below it in the key hierarchy.

Furthermore, as the protected process is now responsible for the trustworthiness of processes it runs, it is no longer necessary to use the seal and unseal functions described above to limit the initiation of processes to certain states of the platform. This is because previous history of the protected process is irrelevant to the protections provided by the protected process.

Although in the foregoing, only one dynamic root key has been illustrated as part of the key hierarchy 12, it will be appreciated that multiple dynamic keys can be included in the hierarchy. Whether there is one or multiple dynamic root keys, the TPM 10 is preferably able to reliably indicate the root key that is currently active. This can be done, for example,

by having the TPM 10 sign a value associated with the current root key, using a TPM identity key. The value that is signed could be the authorisation value of the dynamic root key but is, preferably, a digest of the authorisation value of the root key concerned.

It will be appreciated that many variants are possible to the above described embodiments of the invention. Thus, although in the foregoing description, the dynamic root key 16 is not made available in clear outside of the TPM, it would be possible to provide for the key 16 to be passed to the protected process for use by it, though this is not preferred.

CLAIMS

1. A tree-structured key hierarchy with multiple nodes serving as root nodes dividing the hierarchy into different parts only accessible from corresponding root nodes.

2. Processing apparatus comprising a key-handling unit for handling a tree-structured key hierarchy, the key-handling unit being arranged to treat a selected node of the hierarchy as the current root node such that those parts of the hierarchy that can only be reached by ascent from the current root node are inaccessible, the key-handling unit including an arrangement for changing the node of the hierarchy serving as said current root node.

3. Processing apparatus according to claim 2, wherein the arrangement for changing the current root node is enabled to do so only upon a predetermined set of at least one condition being met.

4. Processing apparatus according to claim 3, wherein at least one predetermined condition comprises the receipt of an authorisation value indicative of digital data.

5. Processing apparatus according to claim 4, wherein said authorisation value is a digest of a protected process associated with the node that is intended to be the new current root node

6. Processing apparatus according to claim 4 or claim 5, wherein at least one predetermined condition comprises that a protected process associated with the node that is intended to be the new current root node is about to be run by the apparatus.

7. Processing apparatus according to claim 6, wherein at least one predetermined condition comprises that any other currently-activated processes running on the apparatus are benign.

8. Apparatus according to any one of claims 3 to 6, wherein at least one predetermined condition comprises that the key-handling apparatus is requested to change the current root node by a root of trust of the apparatus.

9. Processing apparatus according to any one of claims 2 to 8, wherein the node at the head of the hierarchy as judged without regard to which node is the current root node, forms said current root node upon start of the apparatus.

10. Processing apparatus according to any one of claims 2 to 9, wherein the key-handling unit is arranged to hold the current root node internally in unencrypted form at least whilst it remains the current root node.

11. Processing apparatus according to any one of claims 2 to 10, wherein the key-handling unit is arranged always to hold the node at the head of the hierarchy, as judged without regard to which node is the current root node, internally in unencrypted form.

12. Processing apparatus according to any one of claims 2 to 10, wherein the key-handling unit is a Trusted Platform Module according to the TCPA architecture.

13. Processing apparatus according to any one of claims 2 to 11, wherein the key-handling unit is arranged to indicate the current root node by signing a value associated with the node using an identity key associated with the key-handling unit.

14. Processing apparatus according to claim 13 when dependent on claim 4 or claim 5, wherein said value associated with the current root node is the authorisation value associated with the node or a digest of that value.

15. Processing apparatus according to any one of claims 2 to 14, wherein the key-handling unit is so arranged that only a particular type of key node, herein a dynamic key node, can be used as the current root node in addition to the node at the head of the hierarchy as judged without regard to which node is the current root node.

16. Processing apparatus according to claim 15, wherein the key-handling apparatus is arranged, upon receipt of a corresponding command, to generate a dynamic root node as a node of said key hierarchy.

ABSTRACT

Key Hierarchy and Method and Apparatus for Handling the Same

Processing apparatus, such as a trusted platform, is provided with a key-handling unit (10) for handling a tree-structured key hierarchy (12). The key-handling unit (10) is arranged to treat a selected node (16) of the hierarchy as the current root node such that those parts of the hierarchy (12) that can only be reached by ascent from the current root node (16) are inaccessible. The key-handling unit (10) includes an arrangement for changing the node of the hierarchy that serves as the current root node. Also provided is a tree-structured key hierarchy with multiple nodes (11,16) serving as root nodes dividing the hierarchy into different parts only accessible from corresponding root nodes.

(Figure 3)

1/3

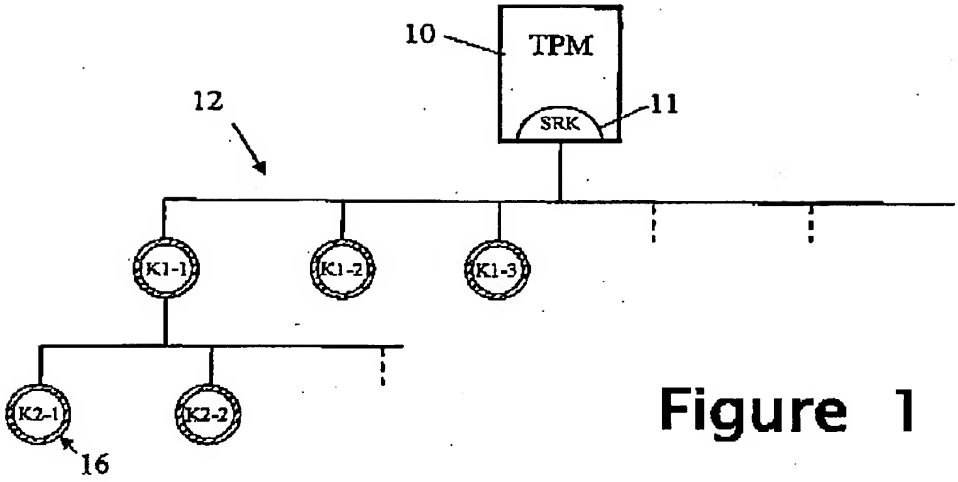

Figure 1

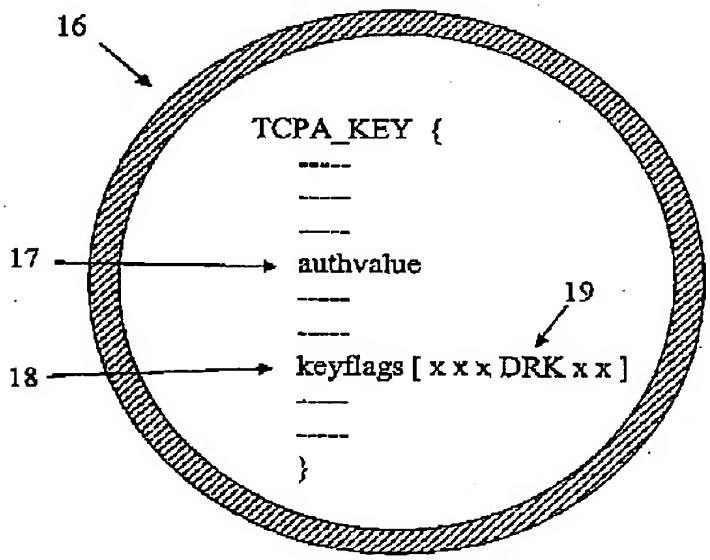

Figure 2

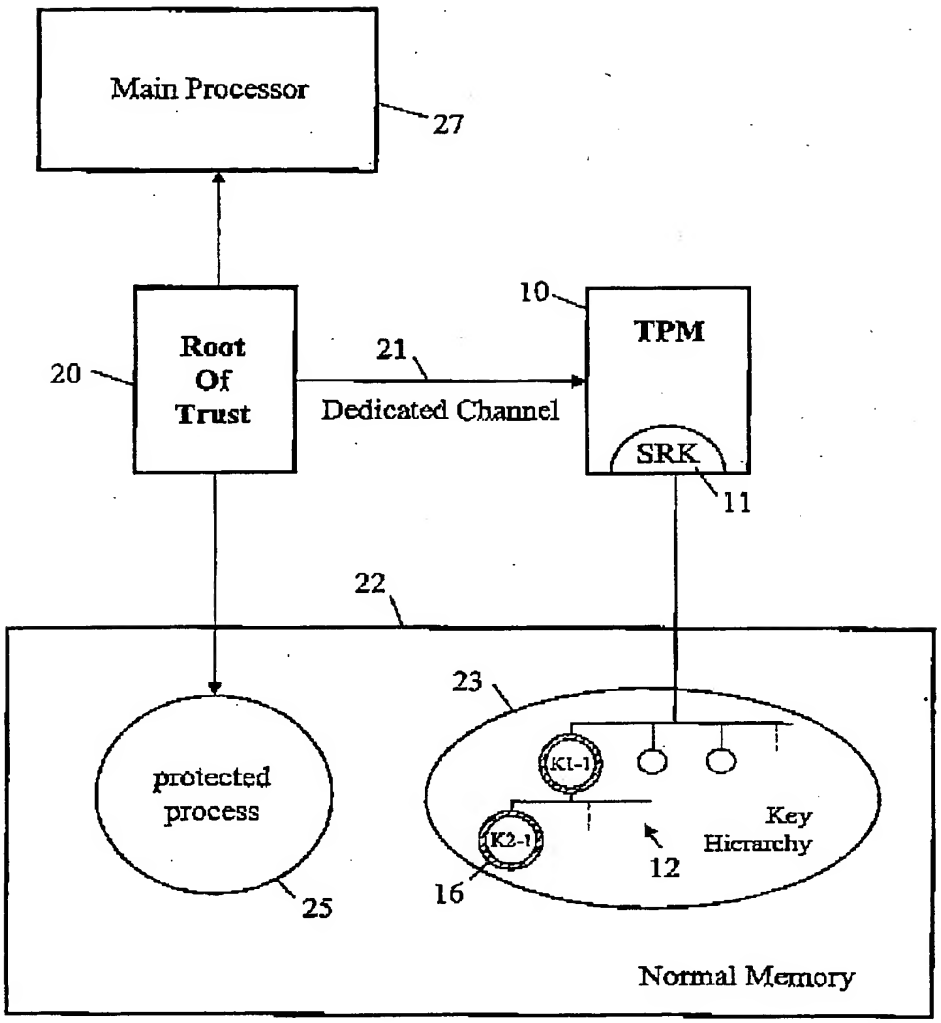
Main Processor
27
Root Of Trust
20
21
Dedicated Channel
10
TPM
SRK
11
22
23
protected process
25
K1-1
K2-1
16
Key Hierarchy
12
Normal Memory

Figure 3

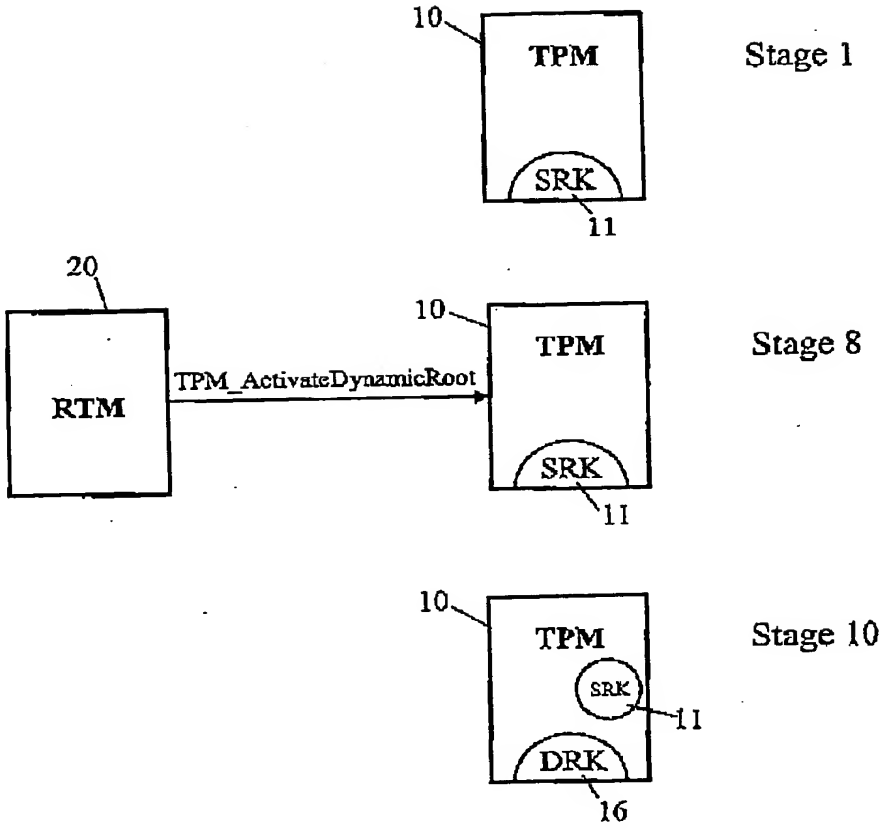
10
TPM
SRK
11
Stage 1
20
RTM
TPM_ActivateDynamicRoot
10
TPM
SRK
11
Stage 8
10
TPM
SRK
11
DRK
16
Stage 10

Figure 4